



TITLE:

2段階アルゴリズムによるSVMの解法 (数理最適化から見た「凸性の深み,非凸性の魅惑」)

AUTHOR(S):

力徳, 正輝; 平井, 広志; 室田, 一雄

---

CITATION:

力徳, 正輝 ...[et al]. 2段階アルゴリズムによるSVMの解法 (数理最適化から見た「凸性の深み,非凸性の魅惑」). 数理解析研究所講究録 2004, 1349: 125-141

ISSUE DATE:

2004-01

URL:

<http://hdl.handle.net/2433/24865>

RIGHT:

## 2 段階アルゴリズムによる SVM の解法

ジャストシステム ナレッジ研究開発部      力徳 正輝 (Masaki Rikitoku)  
Justsystem Knowledge Research Department R&D Division  
東京大学大学院情報理工学系研究科      平井 広志 (Hiroshi Hirai)  
東京大学大学院情報理工学系研究科      室田 一雄 (Kazuo Murota)  
Department of Mathematical Informatics, University of Tokyo

### 1 はじめに

Support Vector Machine(SVM) は与えられたデータを 2 クラスに分類するパターン識別器であり, 現在最も識別性能が高い分類機の 1 つとして知られている [4]. SVM は識別能力を訓練データからの学習によって獲得する. 学習は凸 2 次計画問題 (QP 問題) を解くことに帰着される. 一般にこの QP 問題の係数行列は疎性を持たず, また行列のサイズが訓練データサイズに等しい. したがって, 訓練データサイズが大きい場合には, ハードウェア上の制約により, 係数行列を主記憶に保持することが困難になり, QP 問題に対する汎用アルゴリズムを直接に適用することは困難もしくは非効率的である. そのため SVM により特化した解法の研究が発展してきた [5],[8],[11],[13].

Osuna-Freund-Giros [15] は, より小さい部分 QP 問題を順次解くことによって全体問題の最適解を得る分解アルゴリズムを提案し, このアルゴリズムが QP 問題の汎用アルゴリズムよりも高速であることを実験結果によって示した. この分解アルゴリズムは現在主流となっている SVM の解法の基本的な考え方となっている. Platt [16] は, Osuna らの分解アルゴリズムにおいて部分問題を 2 変数部分問題とする Sequential Minimal Optimization(SMO) アルゴリズムを提案した. 2 変数部分問題は解析的に解くことができ, 1 反復当たりの計算量が少ない. そのため全体として高速な学習アルゴリズムとなる. この手法は現在 SVM の標準的解法となっており, 多くの SVM に実装されている.

SMO は高速である一方で, 1 反復当たり 2 変数しか更新することができず, 最適解を得るために多くの反復数を必要とするため, 丸め誤差等の蓄積によって解の精度が低下することが考えられる. この点を改善するために, 本論文では, SVM の QP 問題に対して, 前半を SMO, 後半を準ニュートン射影法とする 2 段階アルゴリズムを提案する. パターン識別

器としての問題設定から、最適解の近傍においては、変数のほとんどが 0 であり、少数の 0 でない変数 (support vector に対応) と少数の上限値に値をとる変数 (bounded support vector に対応) になる。そのため最適解の近傍では準ニュートン射影法の適用が可能になり、局所的に速い収束と少ない反復数を実現できる。このような 2 段階アルゴリズムは、ポートフォリオ最適化における平均・分散モデルから生じる QP 問題に対して Kawadai-Konno [9] により提案されている。平均・分散モデルは SVM とほぼ同じ QP 問題であるために SVM への適用が可能である。

2 段階アルゴリズムと SMO との比較のために UCI Adult データセットと Web データセット [1] による数値実験を行った。2 段階アルゴリズムの計算時間に関して、通常のパラメータ設定の適用では SMO とほぼ同程度であるが、 $C = 100, 1000$  や最適解の許容誤差  $\epsilon = 10^{-5}, 10^{-6}$  とした場合には SMO の 1.5 倍から 3.0 倍高速になることを確認した。また、準ニュートン射影法の特性によって、すべての設定において反復数の削減を実現しており、最適解への丸め誤差の混入を少なくすることが可能になった。

本論文は次のように構成される。2 章では SVM 学習の 2 次計画問題としての定式化と SMO アルゴリズムの簡単な説明をする。3 章では本論文で提案する 2 段階アルゴリズムの定式化を行う。4 章では SMO と 2 段階アルゴリズムとの比較実験結果を示す。

## 2 Support Vector Machine(SVM)

### 2.1 SVM の凸 2 次計画問題

入力されたデータをある規則によって分類するパターン識別器において、入力データは一般に素性空間と呼ばれるベクトル空間の元として表現される。特に線形識別器はデータ  $\mathbf{x} \in \mathbf{R}^n$  を識別関数

$$y = \text{sgn}(\mathbf{w}^\top \mathbf{x} + b) \quad (2.1)$$

によって  $y = \pm 1$  の 2 値に分類する分類器である。識別関数を決定するパラメータ  $(\mathbf{w}, b)$  は訓練データ  $X = \{(y_i, \mathbf{x}_i) \mid y_i \in \{\pm 1\}, \mathbf{x}_i \in \mathbf{R}^n, i = 1, \dots, l\}$  からの学習によって決定される。

全ての訓練データが超平面  $\mathbf{w}^\top \mathbf{x} + b = 0$  によって 2 値分類可能である

時, 線形分離可能であるという. 線形分離可能である場合には, 学習は

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad (i = 1, \dots, l) \quad (2.2)$$

を満たす  $(\mathbf{w}, b)$  の決定に帰着する. しかし, 一般に訓練データは線形分離可能ではない. また線形分離可能な場合においても, 複数の分離超平面から最適な超平面を決定しなければならない.

この分離超平面の決定に対して, Vapnik-Cortes [3] は, (1)  $1/|\mathbf{w}|$  が最大なものを最適な分離平面とするマージン最大化, (2) 線形分離でない場合や, 例外的なデータが  $y(\mathbf{w}^\top \mathbf{x} + b) < 1$  を満たす領域に侵入することを許すソフトマージン最適化, (3) カーネル関数による非線形識別を可能としたカーネル関数の導入を行い, 現在 Support Vector Machine(SVM) と呼ばれる 2 値分類器を提案し, SVM の学習を QP 問題:

$$\text{最小化} \quad L(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}|\mathbf{w}|^2 + C \sum_{i=1}^l \xi_i \quad (2.3)$$

$$\text{制約} \quad \xi_i = y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) - 1 \geq 0, \quad (i = 1, \dots, l) \quad (2.4)$$

として定式化した. ここで  $C$  はソフトマージン最適化におけるペナルティーを表すパラメータである<sup>1</sup>.

カーネル関数は正定値性

$$\forall N \geq 1, \forall \mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbf{R}^n, \forall c_1, \dots, c_N \in \mathbf{R}, \sum_{i,j=1}^N c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad (2.5)$$

を満たす  $\mathbf{R}^n \times \mathbf{R}^n$  上の対称関数  $K$  として定義される. この正定値性のもとで, カーネル関数には内積表示

$$K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \phi(\mathbf{y}) \quad (2.6)$$

を与える非線形写像  $\phi$  が存在する [7]. カーネル関数としては以下のものが標準的に使用されている.

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y} \quad : \text{線形カーネル}$$

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + r_0)^n, \quad r_0 \in \mathbf{R} \quad : n \text{ 次多項式カーネル}$$

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2}\right), \sigma \in \mathbf{R} \quad : \text{動径基底関数 (RBF) カーネル}$$

<sup>1</sup>1-norm ソフトマージン最適化と呼ばれる定式化

上記の問題 (2.3), (2.4) の双対問題は

$$\text{最小化} \quad W(\alpha) = \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^l \alpha_i \quad (2.7)$$

$$\text{制約} \quad \sum_{i=1}^l y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad (i = 1, \dots, l) \quad (2.8)$$

となる。これに対し, (2.3), (2.4) を主問題と呼ぶ。この QP 問題はカーネル関数によって決まる写像  $\phi$  を陽に含んでおらずカーネル関数  $K$  だけで表現されている。そのため  $\phi$  の像空間の次元が高い場合には, 主問題の変数の個数は大きくなるのに対して, 双対問題の変数の個数は訓練データサイズに抑えられる点で有利である。そのため, SVM の学習においては双対問題を解くことが標準的である。

ここで

$$\partial_i W := \frac{\partial W(\alpha)}{\partial \alpha_i} = y_i \left( \sum_{j=1}^l y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right) - 1 \quad (2.9)$$

とすると, 上記の QP 問題に対する最適性条件 (KKT 条件) は

$$\begin{aligned} \partial_i W + y_i \beta &\geq 0 & \text{if } \alpha_i = 0, \\ \partial_i W + y_i \beta &= 0 & \text{if } 0 < \alpha_i < C, \\ \partial_i W + y_i \beta &\leq 0 & \text{if } \alpha_i = C \end{aligned} \quad (2.10)$$

となる。 $\beta$  は (2.8) の等式制約式に対応する Lagrange 乗数であり, 最適解では識別関数 (2.1) のパラメータ  $b$  と一致する。

学習過程において,  $\beta$  は

$$\beta = -\frac{1}{|I_1|} \sum_{i \in I_1} y_i \partial_i W \quad (2.11)$$

によって推定される<sup>2</sup> [8]。ここで, 添字集合  $I_0, I_1, I_2 \subset \mathbf{Z}_{\geq 0}$  を

$$I_0 = \{i \mid \alpha_i = 0\}, \quad I_1 = \{i \mid 0 < \alpha_i < C\}, \quad I_2 = \{i \mid \alpha_i = C\} \quad (2.12)$$

<sup>2</sup>[10] では  $\beta$  を直接使わない KKT 条件の表現が得られているおり, SMO には標準的に使用されている。

によって定義する.

$0 < \alpha_i < C$  に対応するデータ  $\mathbf{x}_i$  は支持ベクトル (support vector), 特に  $\alpha = C$  に対応するものは上限支持ベクトル (bounded support vector) と呼ばれる. また, 変数  $v$  を

$$v := \max (\{-\partial_i W \mid i \in I_0\} \cup \{|\partial W_i| \mid i \in I_1\}) \cup \{\partial_i W \mid i \in I_2\}) \quad (2.13)$$

で定義すると, KKT 条件を満たすことと  $v = 0$  であることが同値である. 以後,  $v$  を KKT 条件の破れ値とよぶ.

## 2.2 Sequential Minimal Optimization(SMO)

Osuna-Freund-Giros [15] の分解アルゴリズムは QP 問題 (2.8), (2.9) に対して部分問題を順次選択して最適化を行うことによって最適解を得る反復解法の 1 つである. 各反復で決定される作業集合 (working set) と呼ばれる添字集合  $B = \{i_1, i_2, \dots, i_q\}$  に対して部分 QP 問題は

$$\text{最小化} \quad W(\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_q}) = \quad (2.14)$$

$$\frac{1}{2} \sum_{i,j \in B} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i \in B} \alpha_i (1 - y_i \sum_{j \notin B} y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j))$$

$$\text{制約} \quad \sum_{i \in B} y_i \alpha_i = - \sum_{j \notin B} y_j \alpha_j, \quad 0 \leq \alpha_i \leq C \quad (i \in B) \quad (2.15)$$

となる. Osuna らは分解アルゴリズムによって生成される点列が最適解に収束することを証明し, 既存のアルゴリズムより高速であることを実験的に示した.

部分問題では, カーネル行列を主記憶上に保持することが可能であり, 一般の QP 問題のアルゴリズムが使用できる. この手法は現在の SVM の学習アルゴリズムの基礎をなす手法であり, SVM<sup>light</sup> [8] 等の実装されている.

Platt [16] は Osuna らの分解アルゴリズムにおける部分問題を 2 変数最適化問題に設定するアルゴリズムを提案した. 等式制約式を満たしながら更新できる最小の変数の個数が 2 であり, 2 変数部分最適化問題は解析的に解が求まることから, 1 反復当たりの処理時間が少なくなる. このために全体での計算コストが抑えられ高速な学習アルゴリズムとなっている.

この手法は Sequential Minimal Optimization(SMO) と呼ばれ, 部分問題の選択に関するヒューリスティック [10] が適用され現在標準的な SVM の解法となっている.

### Algorithm SMO

- Step 0.  $\alpha$  に初期値を設定する.  
 Step 1. 2 変数  $\alpha_i, \alpha_j$  を選択する.  
 Step 2.  $\alpha_i, \alpha_j$  に関する 2 変数部分最適化問題を解く.  
 Step 3.  $\alpha$  と勾配の更新.  
 Step 4. KKT 条件を判定し, 満たせば終了, そうでない場合には Step 1 へ.

## 3 2 段階アルゴリズム

本論文で提案する 2 段階アルゴリズムを記述するため, 準ニュートン射影法を簡単に説明をする. 最後に SVM の解法として 2 段階アルゴリズムを定式化する.

### 3.1 準ニュートン射影法

線形不等式系を制約式に持つ非線形計画問題

$$\begin{array}{ll} \text{最大化} & f(\mathbf{x}) \\ \text{制約} & g_i = \mathbf{a}_i^\top \mathbf{x} + b_i \leq 0, \quad (i = 1, \dots, m) \end{array} \quad (3.1)$$

を考える. ここで,  $\mathbf{x}, \mathbf{a}_i \in \mathbf{R}^n$  とし,  $I$  を  $\mathbf{x}$  での有効制約式の添字集合とする. すなわち  $I = I(\mathbf{x}) = \{i \mid g_i = 0\}$  であり, 以後のこの表式を用いる.

有効制約式の係数行列を  $A_q = (\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \dots, \mathbf{a}_{i_q})^\top$  とすれば, (3.1) の KKT 条件は

$$\lambda^q = -\nabla f(\mathbf{x}_q) A_q^\top (A_q A_q^\top)^{-1} \geq \mathbf{0} \quad (3.2)$$

となる [12]. KKT 条件が満たさない場合,  $\lambda^q$  の負の要素の最大値を KKT 条件の破れ値と呼ぶ.

このような非線形計画問題に対して有効な解法の1つとして準ニュートン射影法が一般的に使用されており, (3.1) に対する準ニュートン射影法のアルゴリズムは次のようになる [6].

**Algorithm PQN (Projected Quasi Newton method)**

**Step 0.**  $x_0$ , 有効制約式の添字集合  $I(x_0)$ , 有効制約式が生成する部分空間への射影行列を  $P_0$  を設定し,  $H_0 = P_0$  とおく.

**Step 1.**  $d_k = -H_k \nabla f(x_k)$  により探索方向ベクトル  $d_k$  を更新.

**Step 2.**  $d_k = 0$  ならば Step 4 へ, そうでない場合には直線探索によりステップ幅  $\alpha_k$  を設定し,  $x_{k+1} = x_k + \alpha_k d_k$  と更新.

**Step 3.** 有効制約式が不変の場合には BFGS 公式により  $H$  行列を更新. 有効制約式が追加された場合には,  $H$  行列を

$$H_{k+1} = H_k - \frac{H_k a_r^\top a_r H_k}{a_r^\top H_k a_r} \quad (3.3)$$

で更新する. ここで  $a_r$  は追加された有効制約式の方法ベクトルである. さらに,  $I$  に  $r$  を追加,  $P$  を更新して Step 1 へ

**Step 4.** KKT 条件を判定し満足していれば終了. そうでない場合には, KKT 条件の破れ値に対応する有効制約式の方法ベクトル  $a_s$  とすれば,  $I$  より  $s$  を削除,  $P$  を更新,  $H$  行列を

$$H_{k+1} = H_k + \frac{P a_s^\top a_s P}{a_s^\top P a_s} \quad (3.4)$$

で更新して Step 1 へ.

**Step 3** における BFGS 公式による  $H$  行列の更新には

$$H_{k+1} = H_k - \frac{H_k y_k s_k^\top + s_k y_k^\top H_k}{s_k^\top y_k} + \left(1 + \frac{y_k^\top H_k y_k}{s_k^\top y_k}\right) \frac{s_k s_k^\top}{s_k^\top y_k} \quad (3.5)$$

が使用される. ここで,

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (3.6)$$

準ニュートン射影法は, 有効制約式が作る部分空間における, 無制約部分最適化問題を順次解くことによって, 制約付最適化問題の最適解を得



る手法である。(3.3), (3.4) による H 行列の更新は, 有効制約式の変化によって更新された部分空間への射影行列を H 行列に取り込むことに相当する. 更新された H 行列が有効制約式の作る部分空間を不変にすることは, (3.3), (3.4) より容易に確かめることができる. BFGS 公式 (3.5) による H 行列の更新は, セカント条件を要請しながら目的関数のヘッセ行列に近似していく過程である [12], [17].

### 3.2 記憶制限付 BFGS 準ニュートン射影法

BFGS 公式 (3.5) による H 行列の更新と探索方向ベクトルの決定は訓練データサイズの行列とベクトルの演算を必要とするため, 記憶容量, 処理コストの両面で大規模問題に適していない. この問題点を解決する工夫として記憶制限付 BFGS 公式が知られている [14]. この公式は,

$$V_k = I - \frac{\mathbf{y}_k \mathbf{s}_k^\top}{\mathbf{s}_k^\top \mathbf{y}_k} \quad (3.7)$$

として,

$$\begin{aligned} H_k = & (V_{k-t} V_{k-t+1} \cdots V_{k-1})^\top P_0 (V_{k-t} V_{k-t+1} \cdots V_{k-1}) \quad (3.8) \\ & + (V_{k-t+1} \cdots V_{k-1})^\top \frac{\mathbf{s}_{k-t} \mathbf{s}_{k-t}^\top}{\mathbf{s}_{k-t}^\top \mathbf{y}_{k-t}} (V_{k-t+1} \cdots V_{k-1}) \\ & + \cdots \\ & + (V_{k-2} V_{k-1})^\top \frac{\mathbf{s}_{k-3} \mathbf{s}_{k-3}^\top}{\mathbf{s}_{k-3}^\top \mathbf{y}_{k-3}} (V_{k-2} V_{k-1}) \\ & + (V_{k-1})^\top \frac{\mathbf{s}_{k-2} \mathbf{s}_{k-2}^\top}{\mathbf{s}_{k-2}^\top \mathbf{y}_{k-2}} (V_{k-1}) + \frac{\mathbf{s}_{k-1} \mathbf{s}_{k-1}^\top}{\mathbf{s}_{k-1}^\top \mathbf{y}_{k-1}} \end{aligned}$$

とするものであり,  $\{\mathbf{s}_k, \mathbf{y}_k\}$  を更新することで H 行列を更新する. ここで,  $t$  は保持する探索方向ベクトルの個数を定めるパラメータであり, 通常は 1~10 に設定される.  $t = 1$  の場合の記憶制限付 BFGS 公式に基づく準ニュートン法は Solenon-Wolfe 型の共役勾配法と等価であることが知られている. 記憶制限付 BFGS 公式では, 探索方向ベクトル  $\mathbf{d}_k$  の計算において,  $2t$  個のベクトルとそれらの内積計算のみが必要とされるので, BFGS 公式に比べて大幅な記憶領域と計算量の削減が見込める.

記憶制限付 BFGS 公式に基づく準ニュートン射影法は、 $H$  行列を陽に計算しないために、有効制約式の更新に伴う射影行列を  $H$  行列に取り込むことができない。そのため有効制約式が変化されるたびに射影行列  $P$  の更新と、 $H_k = P$  とするリスタートを行う。

**Algorithm LMBFGS-PQN (Limited Memory BFGS  
Projected Quasi Newton method)**

**Step 0.**  $x_0$  を初期化し、 $I = I(x_0)$  を設定する。

**Step 1.** 有効制約式が生成する部分空間への射影行列を  $P$  として  $H_0 = P$  とおく。

**Step 2.** (3.9) を利用して

$$d_k = -H_k \nabla f(x_k)$$

より探索方向ベクトル  $d_k$  を更新。

**Step 3.**  $d_k = 0$  ならば Step 5 へ、そうでない場合には直線探索によりステップ幅  $\alpha_k$  を設定し、 $x_{k+1} = x_k + \alpha_k d_k$  と更新。

**Step 4.** 有効制約式が不変の場合には  $s_k, y_k$  を更新して Step 2 へ。有効制約式が追加された場合  $s_k = y_k = 0$  とし、 $I, P$  を更新して Step 1 へ

**Step 5.** 最適性条件を判定し満足していれば終了。そうでなければ KKT 条件の破れ値に対応する有効制約式の添字  $s$  を  $I$  から削除し、 $P$  を更新して Step 1 へ

### 3.3 2段階アルゴリズム

本論文で提案する 2 段階アルゴリズムは QP 問題 (2.7), (2.8) の前半を SMO, 後半を記憶制限付 BFGS 準ニュートン射影法とするアルゴリズムである。

SMO と準ニュートン射影法の切り替え点は KKT 条件の破れ値が予め定めた閾値  $v_t$  以下になった点とする。

**Algorithm 2STAGE**

**Step 0.**  $\alpha$  に初期値を設定し、閾値  $v_t$  を設定する。

**Step 1.** KKT 条件の破れ値が  $v_t$  以下になっていれば Step 3 へ、そうで

なければ Step 2 へ.

Step 2. SMO アルゴリズムによって  $\alpha$  を更新して Step 1 へ.

Step 3. KKT 条件を満たしていれば終了, そうでなければ Step 4 へ.

Step 4. LMBFGS-PQN アルゴリズムにより  $\alpha$  を更新して Step 3 へ.

SVM の QP 問題においては, 制約式 (2.4) が単純であることが特徴であり, 正規直行基底  $\{e_i \mid i = 1, \dots, l\}$  のみで有効制約式の方方向ベクトルを表すことができる. いまある点  $x$  において,  $I(x) = \{i_1, \dots, i_k\}$  とすれば,  $y = (y_1, \dots, y_l)^\top$  と  $\{e_{i_1}, \dots, e_{i_k}\}$  を使って有効制約式が生成する部分空間への射影行列は

$$P_I = \frac{e_0 e_0^\top}{l - k} + \sum_{i \notin I} e_i e_i^\top \quad (3.9)$$

と表される. ここで,  $e_0 = \sum_{i \notin I} y_i e_i$ .

$p \in I$  なる有効制約式の追加, 削除に伴う射影行列の更新は, (3.9) において  $k = k \pm 1$  と  $e_p e_p^\top$  の追加, 削除することに対応し, SVM の学習において少ない計算量で実行できる.

2 段階アルゴリズムの実装において, 記憶制限付 BFGS 公式のパラメータ  $t$  は 1 に設定した.

$$H_k = V_{k-1}^\top P_I V_{k-1} + \frac{s_{k-1} s_{k-1}^\top}{s_{k-1}^\top y_{k-1}}. \quad (3.10)$$

この場合には共役勾配法となるが, 無制約 2 次計画問題において, 共役勾配法は理論上有限回で最適解を与えるという特性があるので  $t = 1$  で十分であると判断した. また, 探索方向ベクトル  $d_k$  による直線探索は

$$\alpha_{k+1} = \alpha_k + t_k d_k, \quad t_k = -\frac{d_k^\top \nabla W(\alpha)}{d_k^\top Q d_k} \quad (3.11)$$

による厳密な直線探索を使用した.

## 4 実験

### 4.1 実験環境

2段階アルゴリズムの評価のために、SMO との比較実験を行った。SMO の実装にはよく使用されている java 版 LIBSVM-2.4 [2] を使用し、2段階アルゴリズムは  $t = 1$  の記憶制限付 BFGS 準ニュートン射影法を LIBSVM-2.4 に追加する形で実装した。SVM の学習アルゴリズムには必須の処理となっているカーネル行列要素のキャッシュ処理は LIBSVM の API を使用し、キャッシュサイズはデフォルトの 40MB とした。準ニュートン射影法に対する shrinking 処理は SMO の処理とほぼ同じ処理を実装した。

SMO と 2 段階アルゴリズムの評価実験には SVM のベンチマークテストとしてよく利用されている Web データセットと UCI Adult データセット [16] を使用して学習処理時間、反復数を測定した。

SVM のカーネルには線形カーネルを使用し、 $C = 1.0$ 、最適値の許容誤差  $\epsilon = 1.0 \times 10^{-3}$  とした。また、KKT 条件の破れ値の閾値  $v_t = 1.0 \times 10^{-2}$  とした。測定に使用した計算機は CPU Intel Celeron 1.2 GHz、メモリ 512MB である。

### 4.2 実験結果

図 1, 図 2 に web-3a データ, adult-3a データによる SMO と 2 段階アルゴリズムの収束速度の測定結果を示す。横軸を反復数, 縦軸を最適解との距離とした。図 1, 図 2, より SMO の収束はほぼ一次収束であることがわかる。また, 2 段階アルゴリズムでは, 準ニュートン射影法に切り替え後は, 速い収束を示し, SMO に比べ最適解までの反復数の削減に成功していることがわかる。

次に表 1, 表 4.2 に Web データセットと UCI Adult データセットを学習した際の SMO と 2 段階アルゴリズムの反復回数と学習処理時間を示す。SMO と比較して, 準ニュートン法は 1 反復の処理時間が大きい。しかし反復数の減少によってほぼ SMO と同じ学習処理時間を実現している。

SMO は基本的に最急降下法に基づくアルゴリズムであり, 最適解に到達するためには無限回の反復を必要とする。そのため最適解の要求精度が高い場合には多くの反復回数が必要となる。一方, 2 段階アルゴリズムでは

表 1: Web データセットでの反復数, 処理時間測定結果

SMO				2STAGE		
size	SVs(BSVs)	steps	time [ms]	SVs(BSVs)	steps	time [ms]
2477	170(47)	5161	1230	167(47)	3342	1863
3470	220(73)	8698	2711	217(73)	5616	3337
4912	277(107)	12475	4690	272(111)	6348	4759
7366	363(167)	37744	12064	359(172)	13512	11187
9888	453(251)	32956	17774	450(251)	15684	19699
17188	720(481)	103225	76662	704(489)	40507	88211
24692	945(709)	146846	167100	939(718)	54941	173505
49749	1655(1426)	235591	957862	1678(1438)	93838	1525738

解の近傍で, QP 問題に対して理論上有限回で最適解が得られる準ニュートン射影法を利用するため, 高い要求精度に対しても反復回数を SMO に比べて少なくすることができる。

表 3 に web-5a データを使い反復回数, 処理時間を測定した結果を示す。2 段階アルゴリズムは SMO に比較して最大で 3 倍の学習処理速度を示している。また, 要求精度に対して頑健で, 通常の許容誤差  $\epsilon = 1.0 \times 10^{-3}$  において真の最適解の十分良い近似値を得ていることを示している。

一般に SVM の  $C$  値を大きい値に設定すると学習処理時間が増大する。SMO と 2 段階アルゴリズムの  $C$  値依存性を, データとして web-4a を使用して測定した結果を表 4 に示す。

表 4 より  $C$  値が大きい時には SMO に比べ 2 段階アルゴリズムは約 1.5 倍高速であることがわかる。 $C$  値が大きい場合には上限支持ベクトル数が少なく解の近傍ではほぼ無制約最適化問題になり, 準ニュートン法の速い収束によって, 学習が高速になっていると考えられる。

表 2: UCI Adult データセットでの測定結果

size	SMO			2STAGE		
	SVs(BSVs)	steps	times [ms]	SVs(BSVs)	steps	times [ms]
1605	588(524)	10969	3717	588(522)	8152	4176
2265	880(805)	34086	7138	879(807)	12172	6094
3185	1166(1079)	25149	13831	1163(1080)	14159	12970
4781	1735(1645)	35416	32058	1731(1644)	23086	28765
6414	2289(2195)	61279	62469	2285(2199)	24688	56033
11220	4013(3910)	87841	230008	4004(3919)	45854	239493
16100	5759(5656)	133806	593668	5756(5663)	594310	599458
22696	8128(8010)	213529	1356277	8127(8021)	87531	1326390
32561	11527(11359)	151638	3018245	11509(11377)	110229	3004457

## 5 おわりに

本論文で提案した 2 段階アルゴリズムは SMO に比べ、解の近傍では速い収束性を示し、反復数も削減することがわかった。また、準ニュートン射影法は 1 反復当たりの計算量が SMO に比べ大きいですが、測定結果から反復数の削減により後半の処理時間はほぼ同じになっていることを確認できた。

また、 $C$  値が大きい場合や要求精度が高い場合には、2 段階アルゴリズムでは SMO に比べ約 1.5～3.0 倍程度高速になっている。

SVM は訓練データサイズが大きい場合には学習処理時間が長くなるのが欠点の 1 つである。2 段階アルゴリズムは通常のパラメータ領域では SMO とほぼ同程度の処理時間であり、この欠点は克服できていない。これは、SVM の学習において前半が処理時間を大半を占めるためであり、学習処理時間の短縮には前半のアルゴリズムの改良が必要である。これは今後の課題としたい。

表 3: 反復回数, 処理時間の許容誤差依存性

$\epsilon$	SMO		2STAGE	
	steps	time [ms]	steps	time [ms]
$1.0 \times 10^{-3}$	32956	18469	15206	19490
$1.0 \times 10^{-4}$	122105	30905	15501	19036
$1.0 \times 10^{-5}$	271900	52641	15501	19103
$1.0 \times 10^{-6}$	419157	73561	15517	20002

表 4: 反復回数, 処理時間の  $C$  値依存性

$C$	SMO		2STAGE	
	steps	time [ms]	steps	time [ms]
0.1	2519	15389	1813	15653
1.0	25149	20218	13881	20996
$1.0 \times 10^1$	328107	44030	144397	42051
$1.0 \times 10^2$	10323589	243264	1412523	152620
$1.0 \times 10^3$	76383386	1649548	16099972	1076869

## 6 謝辞

本研究は東京大学とジャストシステムの共同研究の成果の一部である。企業における基礎研究の意義を理解し、共同研究の機会を与えてくださったジャストシステムの浮川初子氏、植松直也氏、松田寛氏、研究の初期の段階から有益な助言や示唆を与えてくださった東京大学助教授 松井知巳氏、準ニュートン射影法に関して有益な助言や示唆を与えてくださった東京理科大学教授 矢部博氏、数理計画法について有益な議論をしていただいた東京大学大学院情報理工学系研究科数理情報学専攻第2研究室のみなさま、SVMに関して有益な議論していただいたジャストシステムナレッジ研究開発部のみなさまに謝意を表します。

## 参考文献

- [1] UCI Adult, Web data set, <http://research.microsoft.com/jplatt/smo.html>
- [2] C. C. Chang and C. J. Lin: LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [3] C. Cortes and V. Vapnik: Support vector networks, *Machine Learning*, 20:273 – 297, 1995.
- [4] N. Cristianini and J. Shawe-Taylor: *An Introduction to Support Vector Machines (and Other Kernel Based Learning Methods)*, Cambridge University Press, 2000.
- [5] T.-T. Friess, N. Cristianini and C. Campbell: The kernel adatron algorithm: A fast and simple learning procedure for support vector machines, 15th Intl. Conf. Machine Learning, Morgan Kaufman Publishers, p. 188–196, 1998.
- [6] D. Goldfarb: Extention of davidon’s variable metric method to maximization under linear constraints, *SIAM J. of Applied Mathematics*, 17:739–764, 1969
- [7] D. Haussler: Convolution Kernels on Discrete Structures, UC Santa Cruz Technical Report UCS-CRL-99-10, 1999.
- [8] T. Joachims: Making large-scale support vector machine learning practical, in B. Scholkopf, C. Burges, A. Smola, eds., *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, December 1998.
- [9] N. Kawada and H. Konno: Solving large scale mean-variance models with dense non-factorable covariance matrices, *Journal of the Operations Research Society of Japan*, 44:251–260, 2001



- [10] S. S. Keerthi and S. K. Shevade: C. Bhattacharyya, and K. R. K. Murthy: Improvements to Platt's SMO algorithm for SVM classifier design, Technical Report CD-99-14, Dept. of Mechanical and Production Engineering, National University of Singapore, 1999.
- [11] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya and K. R. K. Murthy: A Fast iterative nearest point algorithm for support vector machine classifier design, IEEE Trans. Neural Networks, 11:124–136, 2000.
- [12] 今野浩, 山下浩: 非線形計画法, 日科技連, 1978.
- [13] O. L. Mangasarian and D. R. Musicant: Lagrangian support vector machines, Journal of Machine Learning Research, 1:161–177, 2001.
- [14] J. Nocedal: Updating quasi-Newton matrices with limited storage, Mathematics of Computation, 35:773–782, 1980.
- [15] E. Osuna, R. Freund, and F. Girosi: An improved training algorithm for support vector machines, In Proc. of IEEE Workshop on Neural Networks for Signal Processing, 276–285, 1997.
- [16] J. Platt: Sequential minimal optimization: A fast algorithm for training support vector machines, Technical Report 98-14, Microsoft Research, Redmond, Washington, April 1998.
- [17] 矢部博, 八巻直一: 非線形計画法, 朝倉書店, 1999.

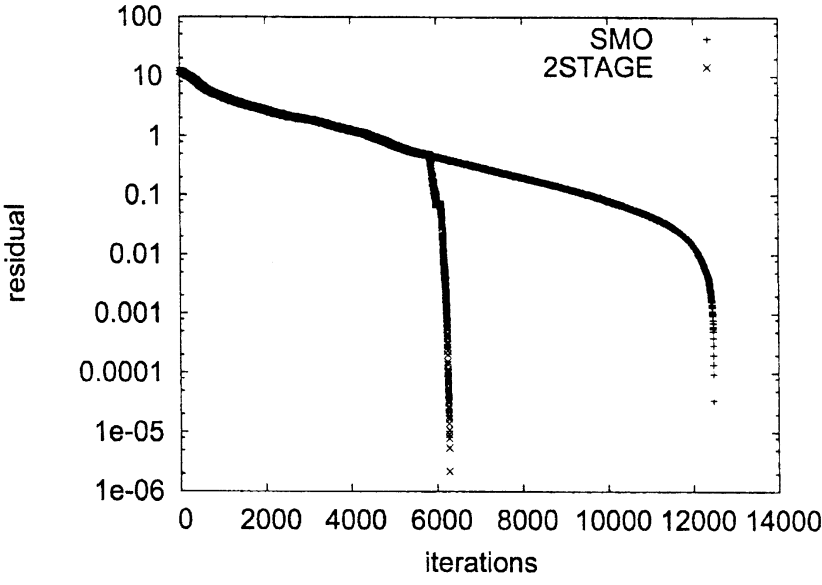


図 1: web-3a に対する収束速度

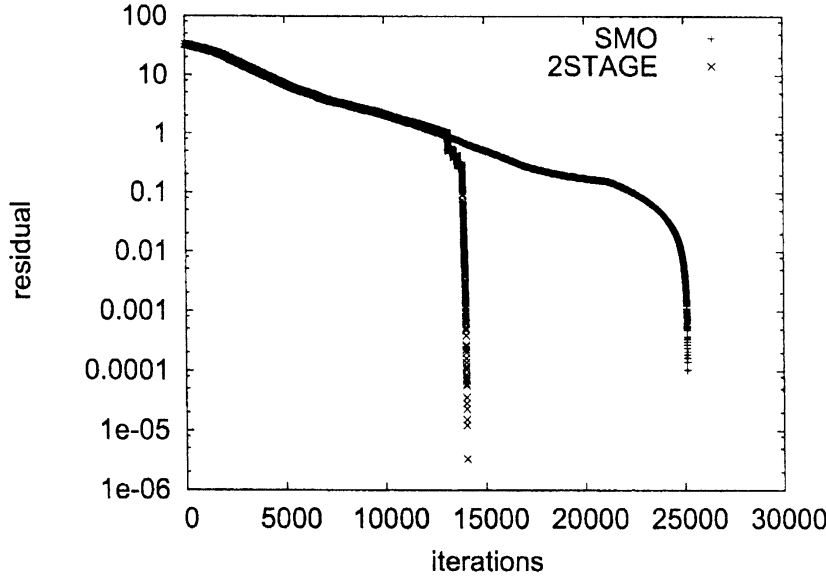


図 2: adult-3a に対する収束速度